AMENDMENTS TO THE CLAIMS

Please amend the claims as follows.

1. (Currently Amended) A network system for key management, comprising:

a server;

a key management system providing process logic for key management system management located on the server;

a key management system storage providing a secure data storage for the key management system;

an application using the key management system to manage an application key; and

an interface providing a means for managing the key management system,

wherein the key management system comprises:

a memory storing data within the key management system;

a hashing module configured to hash~~ing~~ a key encryption key to obtain a key encryption key hash;

an encryption module configured to decrypt~~ing~~ data using the key encryption key and the key encryption key hash; and

a serialization module de-serializing data obtained from the memory, the encryption module, and the serialization module,

wherein the key encryption key comprises a key encryption key PIN, a key encryption key SALT, and a key encryption key ITERATION.

2. (Original) The network system of claim 1, further comprising:

a client computer operatively connected to the server, wherein the client computer comprises a user interface to manage the key management system.

3. (Original) The network system of claim 1, wherein the key management storage is located on the server.

4. (Original) The network system of claim 1, wherein the key management storage is located on a second server operatively connected to the server.

5. (Original) The network system of claim 1, wherein the interface comprises a graphical user interface.

6. (Original) The network system of claim 5, wherein the graphical user interface is integrated into a web browser.

7. (Original) The network system of claim 2, wherein the user interface comprises a graphical user interface.

8. (Original) The network system of claim 7, wherein the graphical user interface is integrated into a web browser.

9. (Original) The network system of claim 2, wherein the client computer and the server are connected using an encrypted connection.

10. (Cancelled)

11. (Currently Amended) A network system for key management, comprising:

a server;

a key management system providing process logic for key management system management located on the server;

a key management system storage providing a secure data storage for the key management system;

an application using the key management system to manage an application key; and

an interface providing a means for managing the key management system,

wherein the key management system comprises:

a memory storing data within the key management system;

a hashing module configured to hash~~ing~~ a key encryption key to obtain a key encryption key hash;

an encryption module configured to decrypt~~ing~~ data and encrypt~~ing~~ data using the key encryption key, the key encryption key hash, and a key decryption key associated with the key encryption key; and

a serialization module de-serializing and serializing data obtained from the memory, the encryption module, and the serialization module, wherein the key encryption key comprises a key encryption key PIN, a key encryption key SALT, and a key encryption key ITERATION.

12. (Previously Presented) The network system of claim 1, wherein the key management system further comprises:
an encoding module for encoding data.

13. (Previously Presented) The network system of claim 1, wherein the hashing module uses an MD5 hashing function.

14. (Previously Presented) The network system of claim 1, wherein the encryption module further comprises a key generation tool.

15. (Previously Presented) The network system of claim 14, wherein the key generation tool comprises a symmetric algorithm.

16. (Previously Presented) The network system of claim 14, wherein the key generation tool comprises an asymmetric algorithm.

17. (Previously Presented) The network system of claim 11, wherein the key management system further comprises:
an encoding module for encoding data.

18. (Previously Presented) The network system of claim 11, wherein the hashing module uses an MD5 hashing function.

19. (Previously Presented) The network system of claim 11, wherein the encryption module further comprises a key generation tool.

20. (Previously Presented) The network system of claim 19, wherein the key generation tool comprises a symmetric algorithm.

21. (Previously Presented) The network system of claim 19, wherein the key generation tool comprises an asymmetric algorithm.

22. (Previously Presented) The network system of claim 1, wherein the interface comprises a means for changing a key encryption key.

23. (Previously Presented) The network system of claim 1, wherein the interface comprises means for starting the key management system.

24. (Previously Presented) The network system of claim 1, wherein the interface comprises means for initializing the key management system.

25. (Previously Presented) The network system of claim 1, wherein the interface comprises means for diagnosing problems with the key management system.

26. (Currently Amended) A network system for key management, comprising:

a server;

a key management system providing process logic for key management system initialization located on the server;

a key management system storage providing a secure data storage for the key management system;

an application using the key management system to manage an application key;

an interface providing a means for inputting data into the key management system; and

a client computer operatively connected to the server, wherein the client computer comprises a user interface to manage the key management system,

wherein the key management system comprises:

a memory storing data within the key management system;

a hashing module configured to hash~~ing~~ a key encryption key to obtain a key encryption key hash;

an encryption module configured to decrypt~~ing~~ data using the key encryption key and the key encryption key hash; and

a serialization module de-serializing data obtained from the memory, the encryption module, and the serialization module,

wherein the key encryption key comprises a key encryption key PIN, a key encryption key SALT, and a key encryption key ITERATION.

27. (Original) A method for retrieving a value secured in a key management system comprising:

receiving a request for the value secured in the key management system;

searching for a key corresponding to the value in a decoded key list; and

retrieving a tuple corresponding to the value, if the key corresponding to the value is in the decoded key list.

28. (Original) The method of claim 27, wherein the key management storage is located on a second server.

29. (Original) The method of claim 27, wherein the key management system interface comprises a graphical user interface.

30. (Original) A method for retrieving a value secured in a key management system comprising:

receiving a request for the value secured in the key management system;

retrieving a serialized file from a key management system storage;

de-serializing the serialized file producing a de-serialized file;

decoding an encoded key list in the de-serialized file to produce a decoded key list;

searching for a key corresponding to the value in the decoded key list;

inputting a key encryption key into the key management system;

hashing the key encryption key to produce a key encryption key hash;

comparing the key encryption key hash to a hashed key encryption key in the de-serialized file;

decrypting a secret token in the de-serialized file using the key encryption key if the key encryption key hash is equal to the hashed key encryption key in the de-serialized file to produce at least one tuple;

storing the at least one tuple in a data structure within the key management system; and

retrieving the tuple corresponding to the value, if the key corresponding to the value is in the decoded key list.

31. (Original) The method of claim 30, further comprising:

searching a local file system, if the key corresponding to the value is not in the decoded key list.

32. (Original) A method for changing an existing key encryption key, comprising:

entering the existing key encryption key;

entering a new key encryption key;

de-serializing a serialized file producing a de-serialized file;

hashing the existing key encryption key producing a hashed key encryption key;

comparing the hashed key encryption key to a key encryption key hash in the de-serialized file;

decrypting a secret token using the existing key encryption key if the hashed key encryption key equals the key encryption key hash producing a tuple;

encrypting the tuple using the new key encryption key producing a new secret token;

hashing the new key encryption key producing a new hashed key encryption key; and

serializing the new hashed key encryption key and the new secret token to produce a new serialized file.

33. (Original) An apparatus for retrieving a value secured in a key management system comprising:

means for receiving a request for the value secured in the key management system;

means for searching for a key corresponding to the value in a decoded key list; and

means for retrieving a tuple corresponding to the value, if the key corresponding to the value is in the decoded key list.

34. A apparatus for retrieving a value secured in a key management system comprising:

means for receiving a request for the value secured in the key management system;

means for retrieving a serialized file from a key management system storage;

means for de-serializing the serialized file producing a de-serialized file;

means for decoding an encoded key list in the de-serialized file to produce a decoded key list;

means for searching for a key corresponding to the value in the decoded key list;

means for inputting a key encryption key into the key management system;

means for hashing the key encryption key to produce a key encryption key hash;

means for comparing the key encryption key hash to a hashed key encryption key in the de-serialized file;

means for decrypting a secret token in the de-serialized file using the key encryption key if the key encryption key hash is equal to the hashed key encryption key in the de-serialized file to produce at least one tuple;

means for storing the at least one tuple in a data structure within the key management system; and

means for retrieving the tuple corresponding to the value, if the key corresponding to the value is in the decoded key list.

35. (Original) An apparatus for changing an existing key encryption key, comprising:

means for entering the existing key encryption key;

means for entering a new key encryption key;

means for de-serializing a serialized file producing a de-serialized file;

means for hashing the existing key encryption key producing a hashed key encryption key;

means for comparing the hashed key encryption key to a key encryption key hash in the de-serialized file;

means for decrypting a secret token using the existing key encryption key if the hashed key encryption key equals the key encryption key hash producing a tuple;

means for encrypting the tuple using the new key encryption key producing a new secret token;

means for hashing the new key encryption key producing a new hashed key encryption key; and

means for serializing the new hashed key encryption key and the new secret token to produce a new serialized file.